

12-13-99

FISH & RICHARDSON P.C.

A

225 Franklin Street
Boston, Massachusetts
02110-2804

Telephone
617 542-5070

Facsimile
617 542-8906

Web Site
www.fr.com

JCS18 U.S. PRO
09/458917
12/10/99

Frederick P. Fish
1855-1930

W.K. Richardson
1859-1951

December 10, 1999

Attorney Docket No.: 07844-353001

Box Patent Application

Assistant Commissioner for Patents
Washington, DC 20231

Presented for filing is a new original patent application of:

Applicant: MARTIN E. NEWELL AND JOHN PETERSON

Title: MANIPULATION OF CURVES AND SURFACES

Enclosed are the following papers, including those required to receive a filing date under 37 CFR 1.53(b):

	<u>Pages</u>
Specification	15
Claims	3
Abstract	1
Declaration	2
Drawing(s)	8

Enclosures:

- Assignment cover sheet and an assignment, 5 pages, and a separate \$40 fee.
- Power of Attorney by Assignee, 2 pages.
- Postcard.

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EL224673055US

I hereby certify under 37 CFR §1 10 that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Assistant Commissioner for Patents, Washington, D C. 20231.

Date of Deposit December 10, 1999

Signature

Stephen Penna

Typed or Printed Name of Person Signing Certificate

12/10/99 JCS18 U.S. PRO

BOSTON

DELAWARE

NEW YORK

SILICON VALLEY

SOUTHERN CALIFORNIA

TWIN CITIES

WASHINGTON, DC

FISH & RICHARDSON P.C.

Assistant Commissioner for Patents

December 10, 1999

Page 2

Basic filing fee	\$760
Total claims in excess of 20 times \$18	\$36
Independent claims in excess of 3 times \$78	\$78
Fee for multiple dependent claims	\$0
Total filing fee:	\$874

A check for the filing fee is enclosed. Please apply any other required fees or any credits to deposit account 06-1050, referencing the attorney docket number shown above.

If this application is found to be incomplete, or if a telephone conference would otherwise be helpful, please call the undersigned at (617) 542-5070.

Kindly acknowledge receipt of this application by returning the enclosed postcard.

Please send all correspondence to:

ROGER S. BOROVOY
Fish & Richardson P.C.
2200 Sand Hill Road, Suite 100
Menlo Park, CA

Respectfully submitted,



David L. Feigenbaum
Reg. No. 30378
Enclosures

APPLICATION
FOR
UNITED STATES LETTERS PATENT

TITLE: MANIPULATION OF CURVES AND SURFACES
APPLICANT: MARTIN E. NEWELL AND JOHN PETERSON

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EL22467305505

I hereby certify under 37 CFR §1.10 that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

Date of Deposit 12-10-99

Signature Stephen Penna

Typed or Printed Name of Person Signing Certificate
Stephen Penna

MANIPULATION OF CURVES AND SURFACES

BACKGROUND

This invention relates to manipulation of curves and surfaces.

When working with artwork in an illustration program like Adobe Illustrator, it is useful to be able to bend or distort curves and surfaces of the artwork in an arbitrary, free-form way. As shown in figure 1, controlled distortion may be used for creative manipulation of type for headings or logos, stretching artwork to fit a particular shape, or giving artwork the appearance of being in a three-dimensional scene without actually defining it in three dimensions.

Bezier curves are widely used to describe curved outlines of objects, such as font glyphs, automobile bodies, and graphic arts curves. A Bezier curve is a mathematical formulation of a curve defined by a sequence of control points. A cubic (3 degree) Bezier curve has four control points. The curve goes through (i.e., the end points of the curve lie on) the first and last control points. A Bezier spline is a sequence of Bezier curves joined end-to-end. A Bezier surface is a mathematical formulation of a surface defined by a rectangular array of control points. A cubic Bezier surface has sixteen control points that all may require manipulation to achieve a desired result.

A known method for manipulating the shapes of curves uses a two-dimensional tensor product surface to specify the distortion. As shown in figure 2, a uniform, rectangular surface is created around the selected artwork. By distorting the shape of the surface, the user may distort the artwork in a corresponding way. The distortion surface is defined by a rectangular grid of control points. Moving the control points of the surface changes the shape in much the same way that moving the control points of a Bézier curve changes the shape of the curve.

Bartels and Beatty in "A Technique for the Direct Manipulation of Spline Curves" (Graphics Interface '89) proposed a scheme in which dragging a point produces limited freedom movement of control points with no freedom on how the curve distorts.

SUMMARY

In general, in one aspect of the invention, in response to relocation information indicative of an intended change in position of a target location on a Bezier shape (e.g., a curve or a surface), new positions are determined for canonical locations on the shape based on predefined intended behaviors of the canonical locations.

Implementations of the invention may include one or more of the following features. The shape may include a d -degree Bezier curve governed by $d+1$ control points and having $d+1$ canonical locations. The control points may be adjusted so that the Bezier shape contains the canonical locations in their new positions. The Bezier shape is rendered based on the new positions of the $d+1$ canonical locations, and the target location in its changed position lies on the rendered Bezier shape.

The predefined intended behavior may be expressed in response functions that define the relationship between changes in positions of target locations and changes in positions of canonical locations. The $d+1$ canonical locations define d sections in order along the shape from one end to the other end, and the predefined intended behavior includes the following: When the target location is in the first section, the one end is relocated, and the other end is constrained to its original location. When the target location is in the d th section, the other end is relocated and the one end is constrained to its original location.

When the Bezier shape is a surface, the position of the target location may be determined by forming a mesh on the surface and searching quadrilaterals of the mesh.

In general, in another aspect of the invention, a user drags a user interface element displayed in association with a Bezier shape to indicate an intended predefined distortion of the Bezier shape. In response to the dragging, the intended predefined distortion is effected by setting new positions for the control points.

Implementations of the invention may include one or more of the following features. The intended predefined distortion is effected by modifying a surface equation to effect the setting of new positions of the control points. The distortion may be symmetric or wave-like. The user interface element may be a handle that is constrained to move in a single direction during dragging.

Among the advantages of the invention are one or more of the following.

Surfaces and curves can be distorted in a more intuitive and controlled way than by manipulating individual control points. Control points may be automatically moved in response to a specified movement of an arbitrary point on a curve. The technique is well behaved no matter which point on the curve is dragged. While some points on the curve can move further than the dragged point, the effect is limited even in the worst case. Boundary conditions can be placed on the distortion of the curve, or surface, to satisfy external

constraints or required behaviors. For example, moving a point near one end of the curve may have no effect on the opposite endpoint.

Other advantages and features will become apparent from the following description and from the claims.

DESCRIPTION

Figure 1 shows distortion examples.

Figure 2 shows distortion components.

Figure 3 illustrates flexible sheet manipulation.

Figure 4 illustrates predefined distortions.

Figure 5 illustrates direct curve manipulation.

Figure 6 shows response curves.

Figure 7 shows quadratic response curves.

Figure 8 illustrates surface dragging.

Figure 9 illustrates sampling to determine surface parameters.

Figure 10 illustrates the treatment of surface drag a series of curve drags.

Figure 11 illustrates a direct manipulation of an "inflate" distortion.

Figure 12 illustrates a wave distortion.

Figure 13 shows linear response curves.

Figure 14 shows scaled response curves.

Figure 15 shows linear response curves.

Figure 16 shows scaled response curves.

Figure 17 shows a process for curve manipulation.

An improved natural user interface for manipulating the contours of curves enables a user simply to select and drag an arbitrary target point on the curve, leaving the software to determine how the rest of the curve should be distorted.

This approach for editing curves forms the basis of a "flexible sheet" style of free-form distortions for surfaces. As shown in figure 3, a distortion surface acts as a flexible sheet that can be pulled and shaped in an intuitive way by clicking and dragging the mouse at any target point on the surface. The software then takes care of manipulating the control points of the surface to provide the flexible sheet illusion.

As shown in figure 4, a different approach provides “canned” distortions that each move in pre-defined, symmetric way. Moving a single control handle takes care of moving all of the control points to create a pre-defined, symmetric distortion.

Curve Manipulation

Because manipulation of curves forms the basis for the “flexible sheet” method of distortion manipulation and because the notation for curves is simpler than for surfaces, we describe it first.

A Bézier curve is defined as

$$\mathbf{C}(t) = \sum_{i=0}^d \mathbf{P}_i B_i^d(t)$$

Where \mathbf{P}_i are the control points, and $B_i^d(t)$ are the Bézier basis functions, defined as

$$B_i^d(t) = \binom{d}{i} t^i (1-t)^{d-i}$$

d is the degree of the curve (e.g., for a cubic curve, $d=3$) and the curve has $d+1$ control points. In order to manipulate the curve, we wish to drag a particular point on the curve $\mathbf{C}(t_{\text{drag}})$ to a point \mathbf{P}_{drag} such that the curve is adjusted in a “natural” way. Doing this requires first determining t_{drag} , which is the parameter of the curve that corresponds to the point on the curve to be moved to the new location. This determination is shown as step 50 in figure 17.

As shown in figure 5, one method for directly manipulating curves developed by Bartels and Beatty is based on the Householder equations. The control points \mathbf{P}_i are modified to create a new set of points according to:

$$\Delta = \mathbf{P}_{\text{drag}} - \mathbf{C}(t_{\text{drag}})$$

$$\hat{\mathbf{P}}_i = \mathbf{P}_i + \Delta \frac{B_i^d(t_{\text{drag}})}{\sum_{i=0}^d (B_i^d(t_{\text{drag}}))^2}$$

(The circumflex ^ is added to curves or points modified by a dragging operation.)

While the Bartels and Beatty method produces a smooth change in the curve, it also produces artifacts. The length of the curve is often preserved, making it difficult to flatten a curved shape. The entire curve segment is adjusted, making local changes difficult. And either the end points are left fixed, producing a large change in the overall curve near the fixed point

when a small change is made, or the endpoints are allowed to move, making it difficult to constrain changes.

Response Curves

Based on a consideration of how the curve should behave when a target point is dragged, the following criteria work well (in the case of a cubic curve) in defining how certain canonical points on the curve should move. An equivalent set can be written for, e.g., the quadratic case.

- For $t_{\text{drag}} > 1/3$, $C(0)$ does not move.
- For $t_{\text{drag}} < 2/3$, $C(1)$ does not move
- Dragging at $t_{\text{drag}} = 0$ behaves just like moving the P_0 control point.
- Dragging at $t_{\text{drag}} = 1$ behaves just like moving the P_3 control point.
- Dragging at $t_{\text{drag}} = 1/3$ or $t_{\text{drag}} = 2/3$ maximally effects that point on the curve.

Given end points A and B (end points A and B are two of the canonical points, and also happen, in this case, to be control points) and intermediate division points X and Y, the three sections of the Bezier are denoted AX (section 1), XY (section 2) and YB (section 3). When dragging a target point located in section 1, end point B will not move, and A, X, and Y will each move based on the position of the target point between A and X. When dragging a target point located in section 2, end points A and B will not move, and X and Y will each move based on the position of the target point between X and Y. When dragging a target point within section 3, end point A will not move, and X, Y, and B will each move based on the position of the target point between Y and B.

Figure 6 shows how these constraints translate into adjustments of the canonical points on the Bezier curve. Each of the four response curves $R_i(t_{\text{drag}})$ shows the extent of motion of one of the canonical points (denoted 0, 1, 2, 3) as a percentage of the dragged position change of any arbitrary drag point along the curve C , where the t_{drag} value of the selected point on the curve is on the horizontal axis, and the vertical axis shows the amount by which the end points and the section dividing points $C(i/3)$, $i = 0, 1, \dots, 3$, on the curve (i.e., the canonical points) are affected by dragging the curve at point $C(t_{\text{drag}})$ to P_{drag} .

Consider the case of dragging the curve at the first end point, $t_{\text{drag}} = 0$. Because (according to the third constraint above) this is the same as dragging P_0 , the response curve R_0 for curve point $C(0)$ has a value of 1 at that point, which means that the point at the first end of the curve (which is the same as the target point) moves as much as the target point is

dragged. The response curve \mathbf{R}_1 for curve point $C(1/3)$ has a value of $8/27$ at $t_{drag} = 0$, because the Bezier basis function for the zeroth control point for a 3 degree curve is $8/27$ evaluated at $t_{drag} = 0$. Thus, when the section dividing point that is $1/3$ of the way along the curve is moved by a unit, the first end point of the curve moves by $8/27$ of a unit. In a similar way, the response curve \mathbf{R}_2 for $C(2/3)$ has a value (of the basis function) of $1/27$ at the first end point of the curve, and the response curve \mathbf{R}_3 for $C(1)$ has a value (of the basis function) of zero at the first end point as required by the constraint above that, for $t < 2/3$, $C(1)$ does not move.

Applying the Method

When a target point $C(t_{drag})$ is dragged to \mathbf{P}_{drag} , the first step is to compute the new positions of the end points and the section dividing points of the new curve $\hat{\mathbf{C}}$ (step 52, figure 17) by applying the response curve to the original curve a

$$\hat{\mathbf{C}}(i/3) = \Delta \mathbf{R}_i(t_{drag}) + \mathbf{C}(i/3), i = 0, \dots, 3$$

Then, the new control points for the curve are determined (step 54) by writing the calculation of the Bézier curve points $\mathbf{C}(i/3)$, $i = 0, \dots, 3$ in matrix form:

$$\mathbf{C}^T = \mathbf{B}\mathbf{P}$$

where \mathbf{B} is the basis coefficient matrix:

$$\mathbf{B} = \begin{bmatrix} B_0^3(0) & B_1^3(0) & B_2^3(0) & B_3^3(0) \\ B_0^3(1/3) & B_1^3(1/3) & B_2^3(1/3) & B_3^3(1/3) \\ B_0^3(2/3) & B_1^3(2/3) & B_2^3(2/3) & B_3^3(2/3) \\ B_0^3(1) & B_1^3(1) & B_2^3(1) & B_3^3(1) \end{bmatrix}$$

To find the new control points, $\hat{\mathbf{C}}(i/t)$ is substituted for $\mathbf{C}(i/t)$ in the equation above, which is then solved for the new control points:

$$\hat{\mathbf{P}} = \hat{\mathbf{C}}\mathbf{B}^{-1}$$

Because the basis coefficient matrix is constant, it can be pre-computed (step 56).

The new curve rendered from the control points (step 58) will move towards \mathbf{P}_{drag} but may not actually reach it. In an interactive program, this is often not noticeable, because the constraint will be satisfied on the next iterations of the mouse tracking loop. However, it is possible to avoid this behavior in a manner described later.

Extensions to Non-cubic Curves

The curve dragging method is extended to non-cubic curves by creating new response curves \mathbf{R}_i for points on the curve at $\mathbf{C}(i/d)$, $i = 0, \dots, d$. Response curves for a quadratic curve, for example, are shown in figure 7. When the curve dragged is part of a multiple-segment Bézier curve, the curve continuity should be maintained by keeping the control points on the adjoining segments co-linear with the ones on the dragged segment, and at the same distance.

Manipulating a Distortion Surface

Extending the methods described above to surfaces provides an intuitive, natural way to edit the distortion surfaces described previously.

A distortion surface of the kind shown in figure 8 is described as:

$$\mathbf{S}(u, v) = \sum_{i=0}^n \sum_{j=0}^m \mathbf{V}_{i,j} B_j^m(u) B_i^n(v)$$

where u, v are the parameters of the surface, $\mathbf{V}_{i,j}$, is the $(n+1) \times (m+1)$ mesh of control points, and

$$B_j^m(u), B_i^n(v)$$

are the Bézier basis functions of degree m, n as described above. The goal of the free-form manipulation is to drag an arbitrary point on the surface $\mathbf{S}(u_{drag}, v_{drag})$ to \mathbf{P}_{drag} in a similar fashion to the curve manipulation method described above.

To drag a point on the surface, the parameters u_{drag}, v_{drag} of the selected point on the surface must be determined, given the selected target point \mathbf{P}_{sel} , so that $\mathbf{S}(u_{drag}, v_{drag})$ is approximately equal to \mathbf{P}_{sel} . As shown in figure 9, to accomplish this, the deformation surface is first coarsely sampled in uniform steps in u and v . A typical sampling rate of $4(n+1) \times 4(m+1)$ is used across the entire surface. By connecting points adjacent to each other in parameter space, the resulting sample points form a mesh of quadrilaterals covering the surface. Each quadrilateral is checked to see if it contains \mathbf{P}_{sel} by checking \mathbf{P}_{sel} against the line equations of the four borders of the quadrilateral (the lines are formed by proceeding around the quadrilateral in a consistent direction, e.g., clockwise). If \mathbf{P}_{sel} is on the same side of all four lines, then it lies inside. If \mathbf{P}_{sel} is found to lie inside one of the quadrilaterals, then the process is repeated, except this time the sampling is done over the parameter range for just that quadrilateral rather than the entire surface. The indices of the sample points containing \mathbf{P}_{sel} are used to determine the parameters u_{drag}, v_{drag} . While this process can be

repeated indefinitely to improve accuracy, in practice two iterations are sufficient for interactive manipulation.

Extending Curve Manipulation to Surfaces

Applying the drag method to the surface proceeds in two steps. First the drag is applied to the iso-curve at v_{drag} . Then, each of the control points of this iso-curve is used as a \mathbf{P}_{drag} for modifying the curves formed by the columns of control points in the surface as shown in figure 10. Because the surface drag is developed as a series of curve drags, it is useful to express the curve manipulation method described in the previous section for dragging the point $\mathbf{C}(t_{drag})$ to \mathbf{P}_{drag} as a function that generates a new set of curve control points from the original control points \mathbf{P} :

$$\hat{\mathbf{P}} = \text{Drag}(\mathbf{P}, t_{drag}, \Delta)$$

where

$$\Delta = \mathbf{P}_{drag} - \mathbf{C}(t_{drag})$$

The iso-curve at v_{drag} \mathbf{C}^v is formed by control points \mathbf{P}^v computed via:

$$\mathbf{P}_j^v = \sum_{i=0}^n \mathbf{V}_{i,j} B_i^n(v_{drag}), j = 0 \dots m$$

A new set of control points for this iso-curve is computed with:

$$\hat{\mathbf{P}}^v = \text{Drag}(\mathbf{P}^v, u_{drag}, \mathbf{P}_{drag} - \mathbf{S}(u_{drag}, v_{drag}))$$

The control points for this new curve are used to drag the columns of the control points by applying them to the surface control points:

for $j = 0 \dots m$

$$\mathbf{P}_i'' = \mathbf{V}_{i,j}, i = 0 \dots n$$

$$\hat{\mathbf{P}}'' = \text{Drag}(\mathbf{P}'', v_{drag}, \hat{\mathbf{P}}^v - \mathbf{P}_j^v)$$

$$\mathbf{V}_{i,j} = \hat{\mathbf{P}}_i'', i = 0 \dots n$$

Manipulating Pre-defined Distortions

We now consider a different approach to surface distortion. The methods described above are useful for free-form manipulation of the distortion surface to produce an arbitrary shape. However, there are many cases where it is useful to produce distortions with a symmetric shape, where the manipulation is confined to a single parameter. Consider an “inflate” pre-defined distortion as shown in figure 11. In this case, as the parameter is changed, the edges of the distortion move in or out. The “inflate” distortion is a bi-quadratic.

The amount of distortion is controlled by a single parameter, which moves the control points on the center of each edge towards (or away from) the center of the mesh. While this manipulation can be done with a traditional user interface tool such as a slider control, another better approach is to provide a handle on the surface that the user drags to the desired shape, subject to the constraints of the symmetry. This closely matches the free-form dragging behavior described above, and makes the surface editing more precise, easier to learn, and easier to use.

To provide this control, a method is needed to convert the movement of the mouse (in one dimension) into a corresponding movement amount for the symmetric movement of the control points in the distortion. This generally involves modifying the surface equation (see above) to incorporate the desired symmetric movement, and then solving that equation for the amount to move the control points.

Continuing with the inflate deform example, a handle is placed at $S(1/2,0)$. This handle is constrained to move only vertically and should track the top of the deformation. To simplify the problem, consider just the top iso-curve of the surface ($v = 0, V_{0,0} \dots 2$). As the handle is dragged, the control point $V_{0,1}$ should move so that the point on the curve $S(1/2,0)$ tracks the mouse as it moves. To find out how far $V_{0,1}$ should move so that the handle tracks the curve, consider the formulation of the curve (with $u = 1/2$, and ignoring v for the moment):

$$S(1/2,0) = \frac{1}{4} V_{0,0} + \frac{1}{2} V_{0,1} + \frac{1}{4} V_{0,2}$$

With $V_{0,0}$ and $V_{0,2}$ fixed, we can see that $V_{0,1}$ moves twice as far as the point on the curve $S(1/2,0)$. Thus, when the handle at $S(1/2,0)$ is dragged, the four control points on the edges are moved twice the distance the handle was moved, giving the illusion that the handle is fixed onto the edge of the distortion.

As a second, more complex example, consider the "wave" distortion shown in figure 12. When the single parameter of this distortion is changed, the left interior control point moves up, and the right interior control point moves down in equal amounts. If y is the wave amount, then the distortion is determined by:

$$V_{1,1} = c + y$$

$$V_{1,2} = c - y$$

where c is the vertical coordinate of the center of the distortion's bounding box. ($V_{i,j}$ refers to just the y coordinate $V_{i,j}$; because the x coordinates are not affected they may be ignored).

To give maximum control, the handle should start out at $S(1/4, 1/2)$. As this handle is dragged up or down, we want it to track the surface, with $V_{1,1}$ moving in the same direction as the handle and $V_{1,2}$ moving in the opposite direction. To do this, we need to find the value corresponding to the difference between the original handle location at $S(1/4, 1/2)$ and P_{drag} . If we call this difference d , and the corresponding distance the control points $V_{1,1}$ and $V_{1,2}$ each move y , we find d is related to y via the equation:

$$\begin{aligned} d &= P_{drag} - S_y(u, v) \\ d &= B_0^2(v) \sum_{j=0}^3 V_{0,j} B_j^3(u) \\ &\quad + B_1^2(v) (V_{1,0} B_0^3(u) + y B_1^3(u) - y B_2^3(u) + V_{1,3} B_3^3(u)) \\ &\quad + B_2^2(v) \sum_{j=0}^3 V_{2,j} B_j^3(u) \end{aligned}$$

Solving for y gives:

$$y = \frac{d - B_0^2(v) \sum_{j=0}^3 V_{0,j} B_j^3(u) - B_2^2(v) \sum_{j=0}^3 V_{2,j} B_j^3(u) - V_{1,0} B_0^3(u) B_1^2(v) + V_{1,3} B_3^3(u) B_1^2(v)}{B_1^2(v) (B_1^3(u) - B_2^3(u))}$$

Collecting terms and substituting $u = 1/4$, $v = 1/2$ gives:

$$y = \frac{256d - 27(V_{0,0} + 2V_{1,0} + V_{2,0}) - 27(V_{0,1} + V_{2,1}) - 9(V_{0,2} + V_{2,2}) - (V_{0,3} + 2V_{1,3} + V_{2,3})}{36}$$

This is the equation relating the mouse movement d to the control point movement y .

Other Deformations

Other direct controls for pre-defined deformations can follow the same general strategy of relating the mouse movement to a point on the surface. Deformations could also have multiple handles (e.g., one for a vertical stretch and another for horizontal) with the implementation strategy being applied to each handle.

Moving a Target Point in a Single Step

As mentioned, the method described above for free-form manipulation of the distortion of the surface using a point on the surface does not adjust a curve sufficiently in a single iteration to move a target point by an amount indicated by dragging. However, the method can be enhanced to move the point by the required amount in a single step. Below

we discuss how this is done for the cubic case. The same approach can be applied to, e.g., the quadratic case.

In the basic method the desired displacements of four canonical points on the curve, at $t = 0, 1/3, 2/3$, and 1 , are specified by the response curves for four specific choices of the dragged point, $t_{drag} = 0, 1/3, 2/3$, and 1 . The behavior of the four on-curve canonical points for other values of t_{drag} was derived by linear interpolation as shown by the linear segments of the response curves. Although linear functions are easy to compute, the result of the choice of linear interpolation is that the dragged point may not be displaced by the correct amount.

Among the possible methods for interpolating behavior for intermediate values of t_{drag} , a set of scaled response curves, S_i derived in a manner explained below, works well. Starting with the response curves, R_i , of the basic method, we find the resulting displacement of the Bezier control points, and of the point, t_{drag} . The point, t_{drag} , will have moved a fraction, f , of the required distance. We therefore scale the displacements of the Bezier control points by the factor $1/f$. The linear nature of the whole system ensures that this will result in the point, t_{drag} , being displaced by the correct amount.

By making different choices of the response curves we can generate different corrected response curves. These different choices will affect the relative amounts by which the various control points move.

The correction may be implemented by literally coding the correction steps as described. Alternatively, the appropriate correction can be derived algebraically, as explained below.

We use the following scaled response curves notation (there has been some reuse of variable names that were previously used for other purposes in the earlier discussion):

$C(t)$ - Position of point on Bezier curve at t
 $\Delta C(t)$ - Change in position of point on Bezier curve at t
 $\Delta CV = [\Delta C(0) \Delta C(1/3) \Delta C(2/3) \Delta C(1)]$ - Vector of curve points at $t = 0, 1/3, 2/3, 1$
 $P_i, i = 0..3$ - Position of Bezier control point i
 $\Delta P_i, i = 0..3$ - Change in position of Bezier control point i
 $P = (P_0 P_1 P_2 P_3)$ - Row vector of P_i
 $\Delta P = (\Delta P_0 \Delta P_1 \Delta P_2 \Delta P_3)$ - Row vector of ΔP_i
 $R_i(t_{\text{drag}}), i = 0..3$ - Value of Response Curve at t_{drag}
 $R(t_{\text{drag}}) = (R_0(t_{\text{drag}}) R_1(t_{\text{drag}}) R_2(t_{\text{drag}}) R_3(t_{\text{drag}}))$ - Row vector of $R_i(t_{\text{drag}})$
 $S_i(t_{\text{drag}}), i = 0..3$ - Value of Scaled Response Curve at t_{drag}
 $S(t_{\text{drag}}) = (S_0(t_{\text{drag}}) S_1(t_{\text{drag}}) S_2(t_{\text{drag}}) S_3(t_{\text{drag}}))$ - Row vector of $S_i(t_{\text{drag}})$
 M - Bezier coefficient matrix
 $T(t) = (1 \ t \ t^2 \ t^3)$ - Row vector of powers of t
 $T(t)^T$ - Transpose of $T(t)$
 A - Power matrix
 ΔL - Vector through which pointing Locator is dragged
 B^n - Vector of Bezier basis functions of degree n

The position of a point, $C(t)$, on a Bezier curve is given by

$$C(t) = P.M.T(t)^T \quad (\text{EQ 1})$$

where

$$M = \begin{bmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The change, $\Delta C(t)$, in position of a point for a given change, ΔP , in the positions of the control points is

$$\Delta C(t) = \Delta P.M.T(t)^T \quad (\text{EQ 2})$$

Applying this to the four canonical points, $t = 0, 1/3, 2/3, 1$

$$[\Delta C(0) \Delta C(1/3) \Delta C(2/3) \Delta C(1)] = \Delta CV = \Delta P.M.A \quad (\text{EQ 3})$$

where

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & \frac{1}{3} & \frac{2}{3} & 1 \\ 0 & \frac{1}{9} & \frac{4}{9} & 1 \\ 0 & \frac{1}{27} & \frac{8}{27} & 1 \end{bmatrix} \quad (\text{EQ 4})$$

is obtained by substituting the values $t = 0, 1/3, 2/3, 1$ into the column $T(t)^T$, expressed as a matrix.

From equation 3, the change in position of the control points required to achieve a change in position of the four canonical points on the curve is

$$\Delta P = \Delta CV \cdot A^{-1} \cdot M^{-1} \quad (\text{EQ 5})$$

where

$$A^{-1} = \begin{bmatrix} 1 & -\frac{11}{2} & 9 & -\frac{9}{2} \\ 0 & 9 & -\frac{45}{2} & \frac{27}{2} \\ 0 & -\frac{9}{2} & 18 & -\frac{27}{2} \\ 0 & 1 & -\frac{9}{2} & \frac{9}{2} \end{bmatrix}$$

and

$$M^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & \frac{1}{3} & \frac{2}{3} & 1 \\ 0 & 0 & \frac{1}{3} & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The changes in position of the canonical curve points, ΔCV , is determined from the response curves

$$\Delta CV = \Delta L \cdot R(t_{\text{drag}}) \quad (\text{EQ 6})$$

Substituting into equation 5

$$\Delta P = \Delta L \cdot R(t_{\text{drag}}) \cdot A^{-1} \cdot M^{-1} \quad (\text{EQ 7})$$

This is essentially the formulation used in the basic method. Now we examine how a point, t , on the curve is affected.

Combining equation 2 and equation 7

$$\Delta C(t) = \Delta L \cdot R(t_{\text{drag}}) \cdot A^{-1} \cdot T(t)^T \quad (\text{EQ 8})$$

This shows that the displacement, $\Delta C(t)$, of the point t on the curve is a scale factor, $R(t_{\text{drag}}) \cdot A^{-1} \cdot T(t)^T$, times the movement, ΔL , of the locator. In particular

$$\Delta C(t_{\text{drag}}) = \Delta L \cdot R(t_{\text{drag}}) \cdot A^{-1} \cdot T(t_{\text{drag}})^T \quad (\text{EQ 9})$$

To ensure that the $C(t_{drag})$ stays with the locator we require $\Delta C(t_{drag}) = \Delta L$. We therefore need to modify the scale factor by $1/(R(t_{drag}) \cdot A^{-1} \cdot T(t_{drag})^T)$. This can be achieved by scaling the response curves to give the scaled response curves, $S(t)$, where

$$S(t_{drag}) = R(t_{drag}) / (R(t_{drag}) \cdot A^{-1} \cdot T(t_{drag})^T) \quad (EQ 10)$$

The revised expression for computing changes, ΔP , in the positions of the Bezier control points for a given displacement, ΔL , of the locator at point t_{drag} on the curve is, from equation 7:

$$\Delta P = \Delta L \cdot S(t_{drag}) \cdot A^{-1} \cdot M^{-1} \quad (EQ 11)$$

The choice of $R(t_{drag})$ used in the basic method is shown in Figure 13. The corresponding scaled response curves are shown in figure 14.

Alternative Response Curves

While the curves of figure 13 work well in an interactive environment, the discontinuity of slope in $S_1(t_{drag})$ at $t_{drag} = 1/3$, and similarly in $S_2(t_{drag})$, may be considered undesirable. This can be fixed by making a different choice of $R_1(t_{drag})$ and $R_2(t_{drag})$. For example, figure 15 shows, for the case of a two-degree curve) response curves in which $R_1(t_{drag})$ and $R_2(t_{drag})$ are 2nd degree polynomials interpolating the same boundary conditions:

$$R_1(t_{drag}) = 8/27 + t \cdot 89/27 + t^2 \cdot 32/9$$

$$R_2(t_{drag}) = R_1(1 - t_{drag})$$

The corresponding scaled response curves are shown in figure 16.

Application to Curves of Other Degree

The treatment given here may be applied directly to Bezier curves of degree other than three. Differences include the size of the various vectors and matrices, the coefficients in the Bezier coefficient matrix, M , and the boundary conditions for the response curves. For degree 2 curves (quadratic), the boundary conditions require that for $t_{drag} < 1/2$, the end of the curve, $C(1)$, does not move, and similarly for $t_{drag} > 1/2$ with respect to end $C(0)$. See figure 7.

The linear case is the same as a linear (degree one) example proposed by Bartels and Beatty.

End Conditions

When the end, $t_{drag} = 0$, of a curve is moved, the treatment given here causes the curve to distort as if only the control point, P_0 , were moved - all other control points remain

fixed. This is not always desirable. For example, the curve may be part of a composite curve, consisting of many Bezier segments connected end-to-end, with slope continuity between segments. In this case, moving the end of a curve segment should move not only the control point, P_0 , but also the adjacent control point P_1 . This will have the effect of maintaining the slope of the end of the curve. If the same treatment is given to the adjoining curve segment, whose end is also being dragged at the same time, then continuity of slope is maintained. This behavior can be incorporated into the present scheme by modifying the boundary conditions on the response curves, as follows:

$$R_1(0) = R_2(1) = B_0^3(1/3) + B_1^3(1/3) = 20/27$$

$$R_1(1) = R_2(0) = B_0^3(2/3) + B_1^3(2/3) = 7/27$$

This gives rise to the following response curves:

$$R_0(t_{\text{drag}}) = \text{if } (t_{\text{drag}} < 1/3) \text{ then } (1 - 3 \cdot t_{\text{drag}}) \text{ else } (0)$$

$$R_1(t_{\text{drag}}) = \text{if } (t_{\text{drag}} < 1/3) \text{ then } (20/27 + t_{\text{drag}} \cdot 21/27) \text{ else } (37/27 - t_{\text{drag}} \cdot 30/27)$$

$$R_2(t_{\text{drag}}) = R_1(1 - t_{\text{drag}})$$

$$R_3(t_{\text{drag}}) = R_0(1 - t_{\text{drag}})$$

Corresponding scaled response curves can be derived from these expressions using equation 10. In any given implementation, both forms of boundary conditions might be used. The original form may be appropriate for ends of composite curves, or for moving interior segment ends where slope continuity is not required. This new set of response curves is more appropriate for interior segment ends where slope continuity is required. The remaining question concerns maintenance of slope continuity between joined Bezier segments when $t_{\text{drag}} > 0$, because the adjacent segment will not normally be affected. Two strategies present themselves. Both involve repositioning the adjacent slope control point of the adjacent segment so that it is co-linear with the end point and slope point of the current segment. One strategy would maintain the length of the adjacent slope vector. The other would maintain the ratio of the magnitudes of the slope vectors at the adjoining ends.

Other embodiments are within the scope of the following claims.

What is claimed is:

1 1. A method comprising
2 receiving relocation information indicative of an intended change in position
3 of a target location on a Bezier shape, the contour of the Bezier shape being governed by
4 control points, and

5 in response to the relocation information, determining new positions for
6 canonical locations on the shape based on predefined intended behaviors of the canonical
7 locations.

1 2. The method of claim 1 in which the shape comprises a d-degree Bezier curve
2 governed by d+1 control points.

3 3. The method of claim 2 in which there are d+1 canonical locations.

4 4. The method of claim 1 further comprising
5 adjusting the control points so that the Bezier shape contains the canonical locations
6 in their new positions.

7 5. The method of claim 1 in which the Bezier shape comprises a curve or a
8 surface.

9 6. The method of claim 1 further comprising rendering the Bezier shape based
10 on the new positions of the d+1 canonical locations.

11 7. The method of claim 1 in which the target location in its changed position lies
12 on the rendered Bezier shape.

1 8. The method of claim 1 in which the predefined intended behavior is expressed
2 in response functions that define the relationship between changes in positions of target
3 locations and changes in positions of canonical locations.

4 9. The method of claim 3 in which the d+1 canonical locations define d sections
5 in order along the shape from one end to the other end, and the predefined intended behavior
6 comprises the following:

7 when the target location is in the first section, the one end is relocated, and the other
8 end is constrained to its original location, and

6 when the target location is in the dth section, the other end is relocated and the one
7 end is constrained to its original location.

1 10. The method of claim 1 in which the Bezier shape comprises a d-degree curve,
2 the one end and the other end comprise end points of the curve, and the target location
3 comprises a point along the curve.

1 11. The method of claim 1 in which the Bezier shape comprises a 3-degree curve
2 and there are four canonical locations.

1 12. The method of claim 1 in which the Bezier shape comprises a 2-degree curve
2 and there are three canonical locations.

1 13. The method of claim 1 in which the control points are adjusted using a pre-
2 computed basis coefficient matrix.

1 14. The method of claim 1 in which the Bezier shape comprises a surface and in
2 which the position of the target location is determined by forming a mesh on the surface and
3 searching quadrilaterals of the mesh.

1 15. The method of claim 11 further comprising processing the relocation
2 information as a series of curve relocations.

1 16. A medium storing machine readable instructions arranged to cause a machine
2 to

3 receive relocation information indicative of an intended change in position of a target
4 location on a Bezier shape, the contour of the Bezier shape being governed by control points,
5 and

6 in response to the relocation information, determine new positions for canonical
7 locations on the shape based on predefined intended behaviors of the canonical locations.

1 17. A method comprising
2 receiving relocation information indicative of an intended change in position of a
3 target location on a Bezier shape, the contour of the Bezier shape being governed by control
4 points,

5 in response to the relocation information, determining new positions for canonical
6 locations on the shape based on predefined intended behaviors of the canonical locations, the
7 predefined intended behaviors being expressed in scaled response functions that define the
8 relationship between changes in positions of target locations and changes in positions of
9 canonical positions,

adjusting the control points so that the Bezier shape contains the canonical locations in their new positions, and

rendering the Bezier shape based on the new positions of the canonical locations so that the target location in its changed position lies on the rendered Bezier shape.

18. A method comprising
enabling a user to drag a user interface element displayed in association with a Bezier shape to indicate an intended predefined distortion of the Bezier shape, the contour of the Bezier shape being governed by control points, and
in response to the dragging, effecting the intended predefined distortion by setting new positions for the control points.

19. The method of claim 18 in which the intended predefined distortion is effected by modifying a surface equation to effect the setting of new positions of the control points.

20. The method of claim 18 in which the distortion is symmetric.

21. The method of claim 18 in which the distortion is wave-like.

22. The method of claim 18 in which the user interface element comprises a handle that is constrained to move in a single direction during dragging.

MANIPULATION OF SURVES AND SURFACES

ABSTRACT

5 In a shape manipulation method, relocation information is received indicative of an intended change in position of a target location on a Bezier shape, the contour of the Bezier shape being governed by control points. In response to the relocation information, new positions are determined for canonical locations on the shape based on predefined intended behaviors of the canonical locations.

20001410.doc

P328



Figure 1

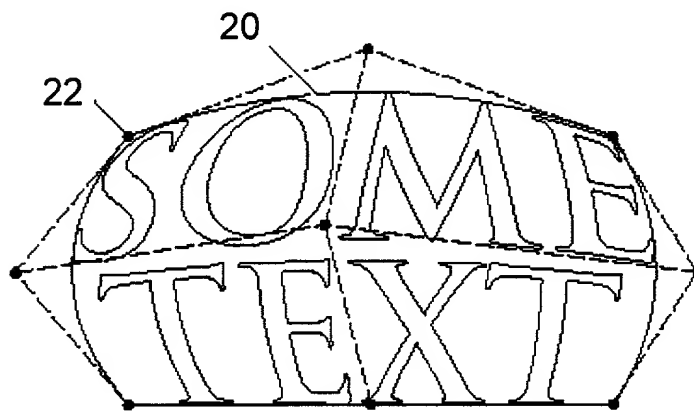


Figure 2

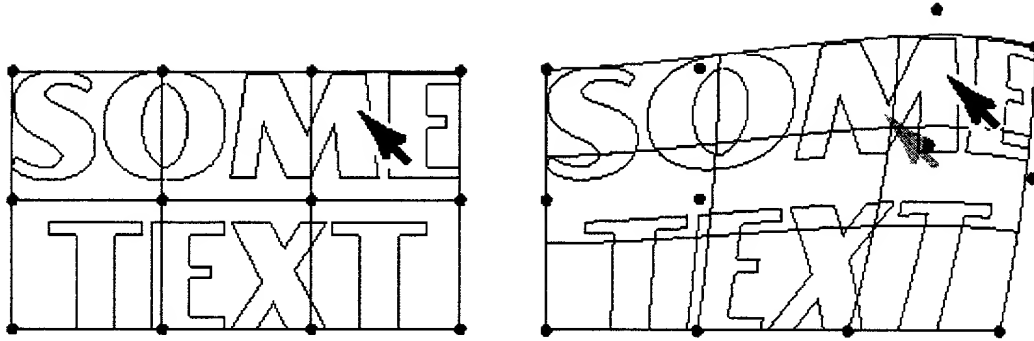


Figure 3

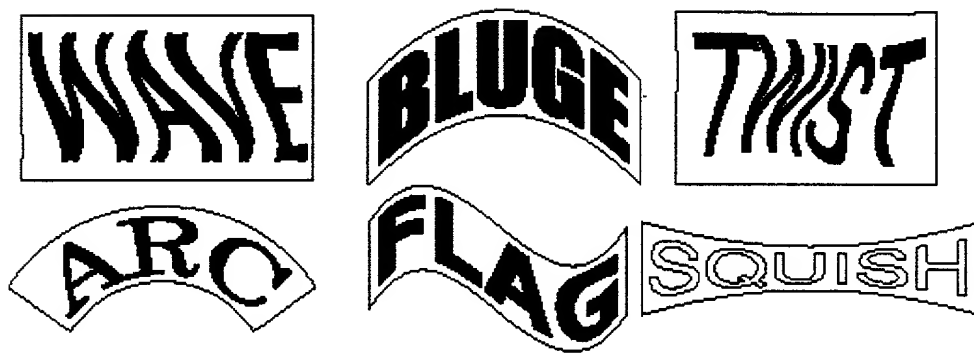


Figure 4

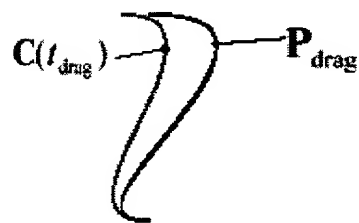


Figure 5

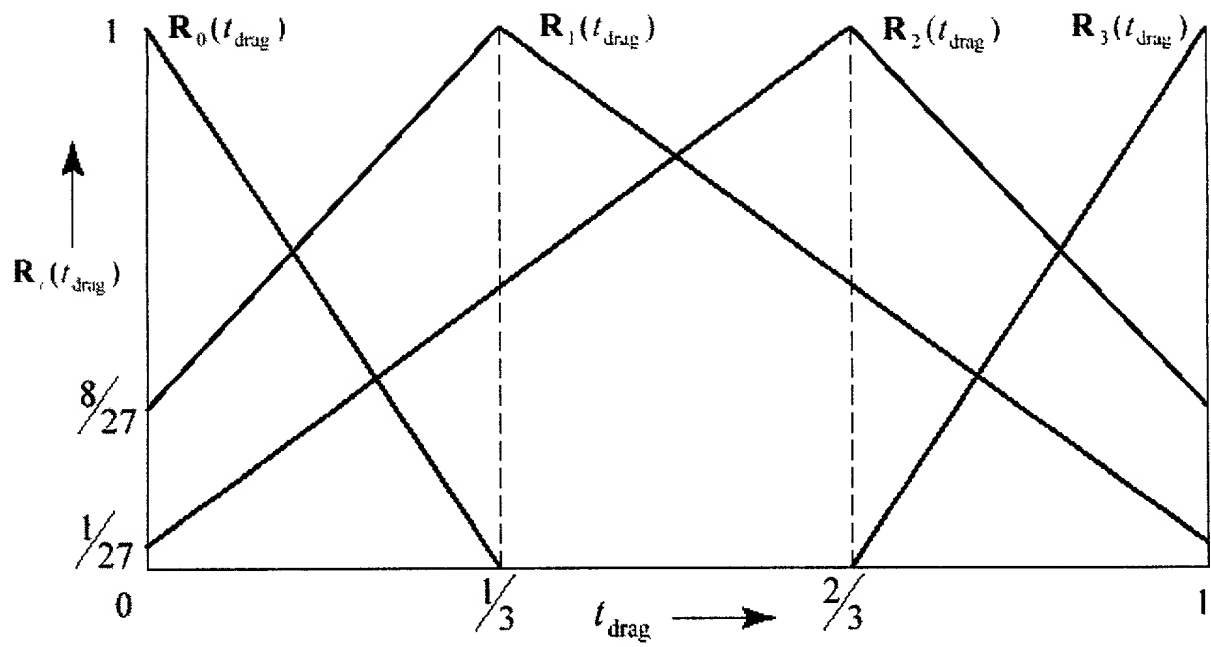


Figure 6

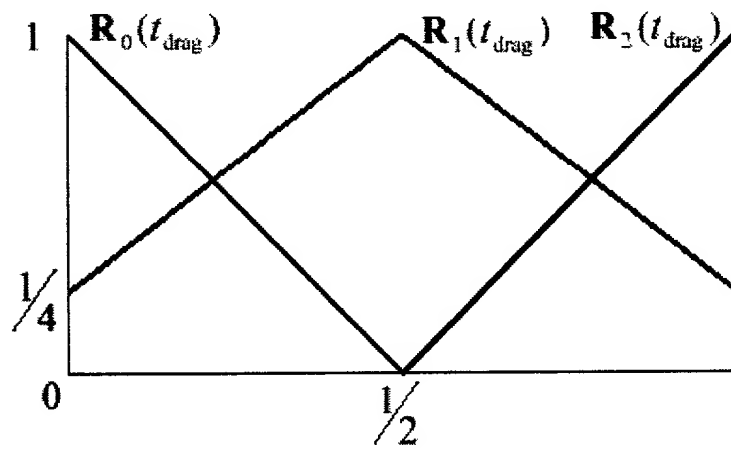


Figure 7

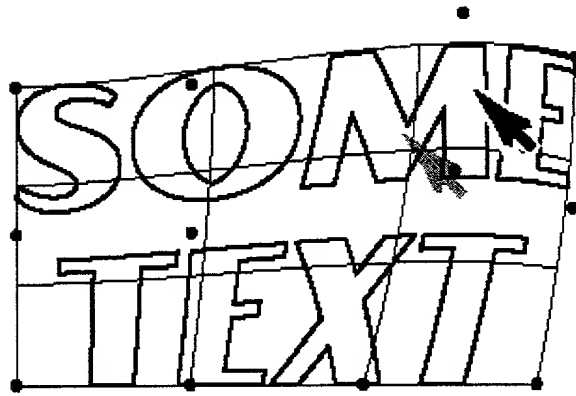


Figure 8

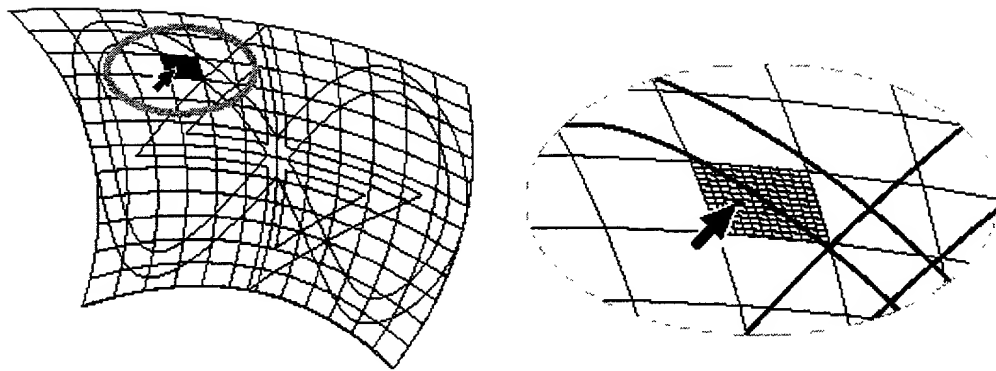


Figure 9

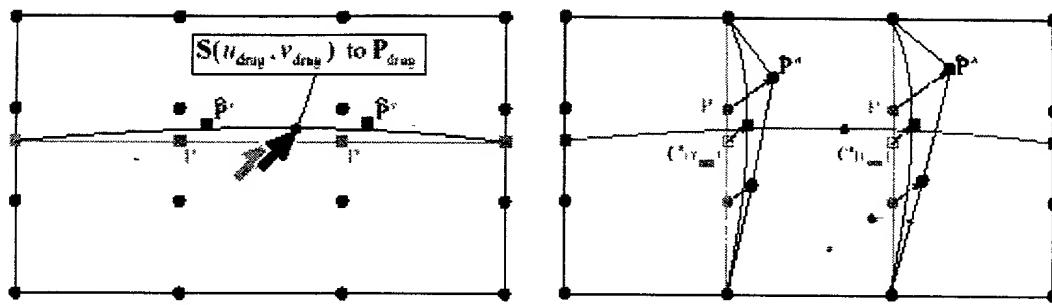


Figure 10

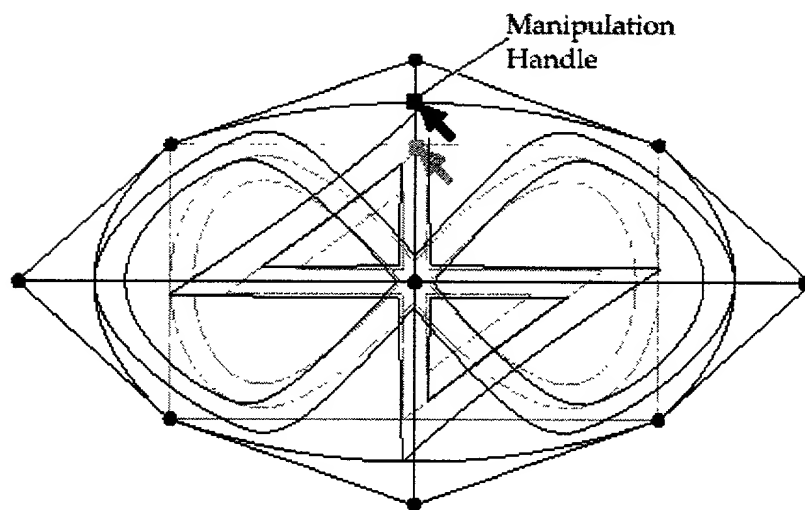


Figure 11

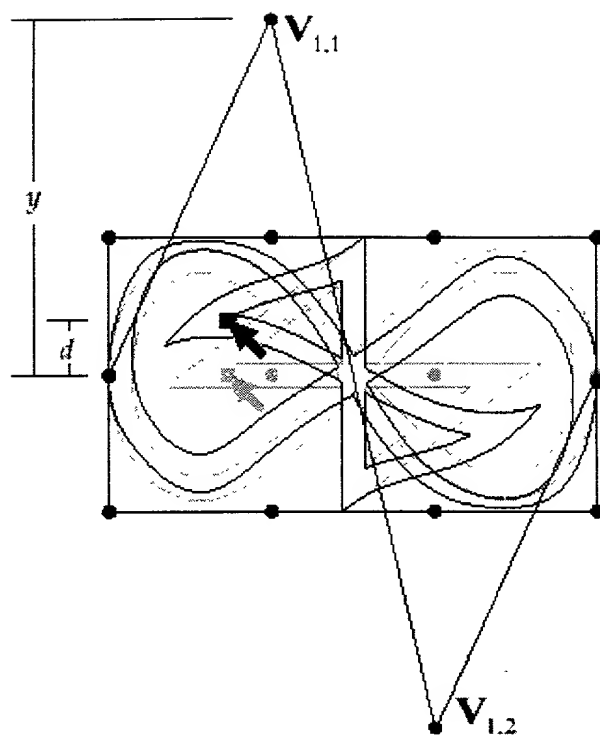


Figure 12

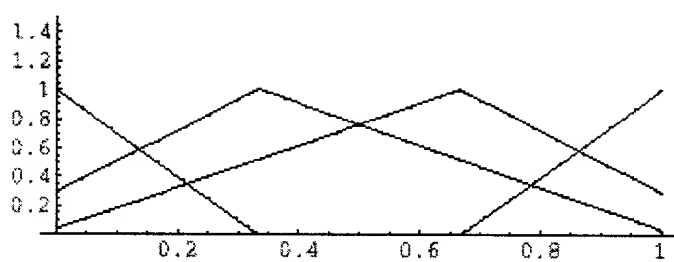


Figure 13

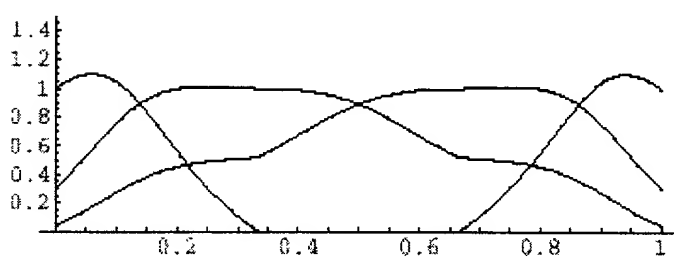


Figure 14

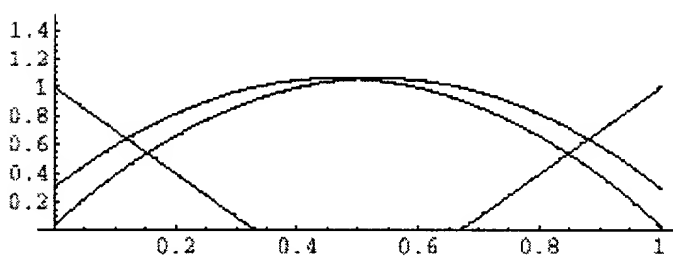


Figure 15

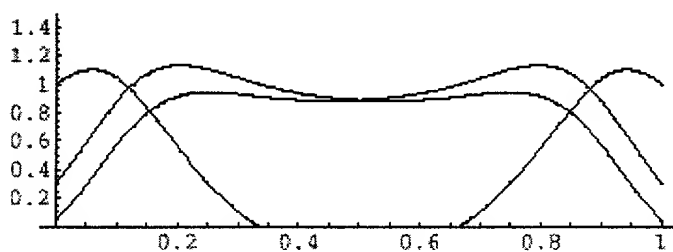


Figure 16

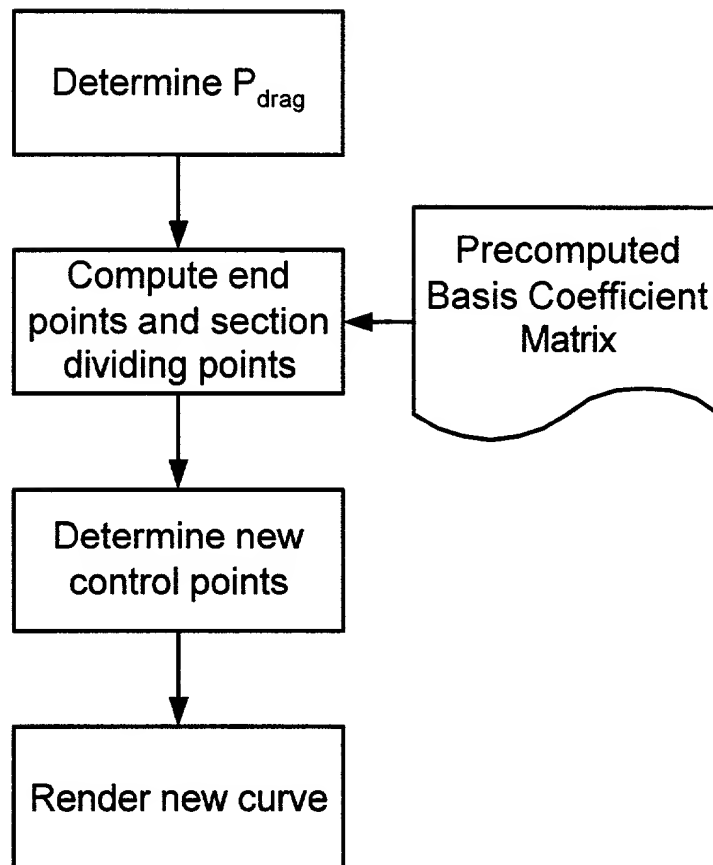


Figure 17

COMBINED DECLARATION AND POWER OF ATTORNEY

As a below-named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

MANIPULATION OF CURVES AND SURFACES

the specification of which:

☒ is attached hereto.

☐ was filed on _____.

☐ under Application No. _____.

☐ with Express Mail No. _____ (Application Number not yet known).

☐ was described and claimed in PCT International Application No. _____.

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, code of Federal Regulations, Section 1.56(a).

I hereby appoint the following attorneys and/or agents to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith, and direct that all correspondence be addressed to that Customer Number:

Customer Number 021876

Address all telephone calls to **David L. Feigenbaum, Esq., Reg. No. 30,378** at telephone number (617) 542-5070.

☒ For Assigned Inventions: I understand that the purpose of making this appointment is to permit prosecution of patent applications for the above-identified invention for the benefit of my assignee, and that this appointment does not create an attorney-client relationship between me and these appointees.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full name of inventor: Martin E. Newell

Inventor's signature

✓ Martin E. Newell

Date:

✓ 8th December, 1999

Residence:

410 Adobe Place, Palo Alto, CA 94306

Citizen of:

U.S.A.

Post Office Address:

410 Adobe Place, Palo Alto, CA 94306

Full name of inventor: John Peterson

Inventor's signature



Date:

18-DEC-99

Residence:

12 Bishop Lane, Menlo Park, CA 94025

Citizen of:

U.S.A.

Post Office Address:

12 Bishop Lane, Menlo Park, CA 94025

12
B
I
S
H
O
P
L
A
N
E
M
E
N
L
O
P
A
R
K
C
A
9
4
0
2
5

POWER OF ATTORNEY BY ASSIGNEE
Attorney Docket No. 07844/353001
Client No. P328

IN THE U.S. PATENT AND TRADEMARK OFFICE

Inventors: Martin E. Newell et al.
Serial No.:
Filed:
For: MANIPULATION OF CURVES AND SURFACES

POWER OF ATTORNEY BY ASSIGNEE AND ELECTION OF ASSIGNEE
TO CONDUCT PROSECUTION TO EXCLUSION OF INVENTOR(S) UNDER 37 C.F.R. 3.71 et seq.

The undersigned, as authorized representative of the assignee of the entire right, title and interest in the above-identified application, hereby appoints the practitioners associated with the Customer Number provided below to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith with full powers of substitution and revocation, said appointment to be to the exclusion of the inventor(s) and his/their attorney in accordance with the provisions of 37 C.F.R. §3.71 et seq. of the Patent Office, and directs that all correspondence be addressed to that Customer Number:

Customer Number 021,876

Ownership is in the assignee, Adobe Systems Incorporated, a Delaware corporation having a place of business at 345 Park Avenue, San Jose, California 95110, by virtue of an assignment from the inventor(s) of the patent application identified above.

- [X] An assignment from the inventor(s) is filed herewith.
- [] An assignment from the inventor(s) to _____ was recorded in the Patent and Trademark Office at Reel _____ Frame _____ on _____.
- [] An assignment from Assignor _____ to Assignee _____ was recorded in the Patent and Trademark Office at Reel _____ Frame _____ on _____.
- [] An assignment from Assignor _____ to Assignee _____ was recorded in the Patent and Trademark Office at Reel _____ Frame _____ on _____.

The undersigned has reviewed all the documents in the chain of title of the patent application identified above and, to the best of undersigned's knowledge and belief, title is in the assignee identified above.

The undersigned, whose title is supplied below, is empowered to act on behalf of the assignee.

"Express Mail" mailing label number EL224673055 US
Date of Deposit 12-10-99

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231

Stephen Penna

POWER OF ATTORNEY BY ASSIGNEE

Attorney Docket No. 07844/353001

Client No. P328

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patents issued thereon.

Signature:

✓ 

Printed name:

Peter F. Ullmann

Title:

Patent Programs Manager

of

Assignee:

Adobe Systems Incorporated